

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №1
по курсу «Программирование графических процессоров»**

Освоение программного обеспечения для работы с технологией CUDA.

Примитивные операции над векторами.

Выполнил: И.Т. Батыновский
Группа: 8О-407Б
Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2021

Условие

Цель работы: Ознакомление и установка программного обеспечения для работы с программно-аппаратной архитектурой параллельных вычислений(CUDA).
Реализация одной из примитивных операций над векторами.

Вариант 2. Вычитание векторов.

Программное и аппаратное обеспечение

Device: GeForce GTX 970

Размер глобальной памяти: 4294967296

Размер константной памяти : 65536

Размер разделяемой памяти: 49152

Регистров на блок: 65536

Максимум потоков на блок: 1024

Количество мультипроцессоров : 13

OS: Windows 10

Редактор: CLion

Метод решения

Поэлементное вычитание каждого элемента с помощью нахождения индекса нити.

```
int index = threadIdx.x + blockIdx.x * blockDim.x;
```

Описание программы

Создаются три вектора в глобальной памяти

```
myType * v1 = new myType[n];
```

```
myType * v2 = new myType[n];
```

```
myType * res = new myType[n];
```

```
cudaMalloc((void **) &d_v1, size);
```

```
cudaMalloc((void **) &d_v2, size);
```

```
cudaMalloc((void **) &d_res, size);
```

```

cudaMemcpy(d_v1, v1, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_v2, v2, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_res, res, size, cudaMemcpyHostToDevice);

```

Запускается ядро, которое находит разность.

```

mykernel<<<(n / THREADS_PER_BLOCK + 1), THREADS_PER_BLOCK>>>(d_v1, d_v2,
d_res, n);

```

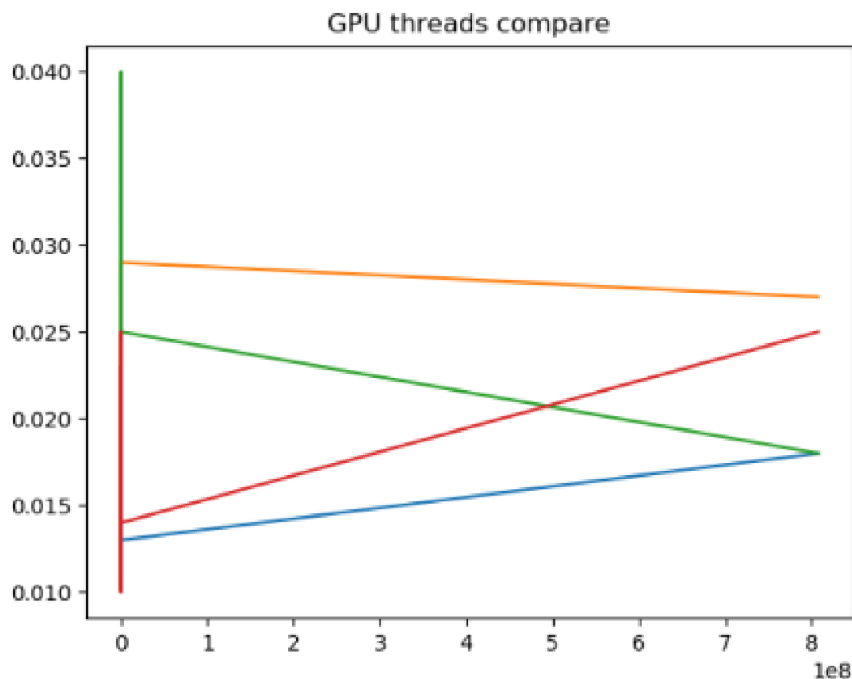
```

__global__ void mykernel(myType * v1, myType * v2, myType * res, unsigned long long edge)
{
    int index = threadIdx.x + blockIdx.x * blockDim.x;
    if (index < edge)
        res[index] = v1[index] - v2[index];
}

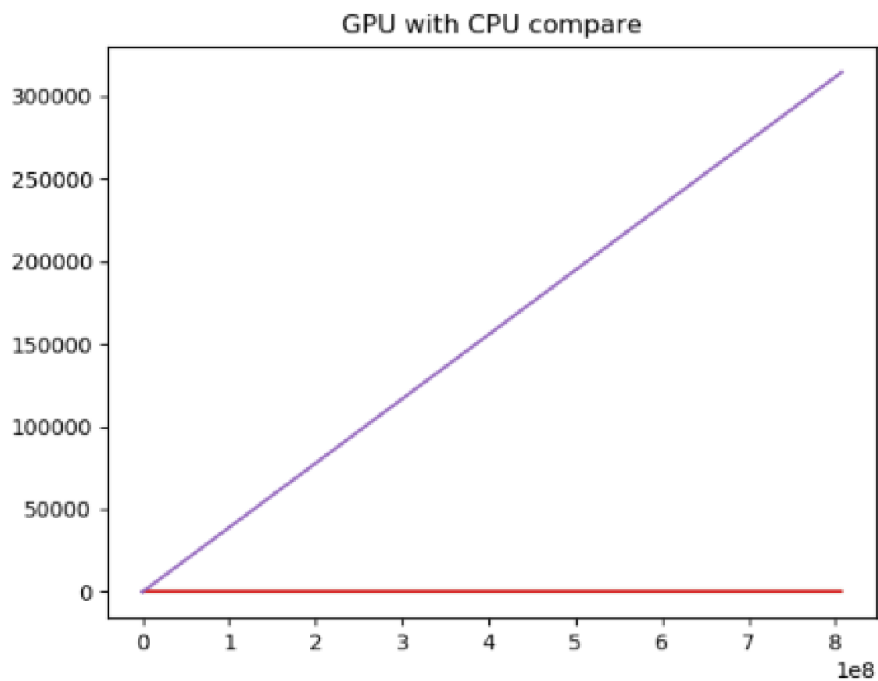
```

Результаты

После тестирования программы на различных данных и с отличающимся количеством блоков и нитей были получены следующие результаты. По оси OX - размер теста, OY - время выполнения.



Также, я провел сравнение времени непосредственных вычислений между CPU и GPU, не беря в учет время копирования векторов с host на gpu. Заметно, что разница колоссальная.



Выводы

Можно также заметить, что способ вычисления значений на CPU, при небольших данных не отличается от GPU, а иногда даже является более быстрым, таким образом преимущество становится заметным.